Java Programming

Section 1 – Building block , I/O, Operators & Expressions

- Variable
- Data Type
- Input and Output
- Initialization Constant
- Escape Sequence
- Arithmetic Operator
- Relational & Logical Operator Increment & Decrement Operator
- Assignment Operator
- Conditional Expression
- Precedence and evaluation order
- Type Conversion
- Using Boolean and char types
- Using Input with Scanner Class

*1. Write a program to print given format using System.out.println() function.

```
*
***
*****
******
*******
```

*2. Write a program to find out simple interest (SI).
*3. Write a program to find gross salary (Hint :-GS=BS+DA+TA).
*4. Write a program for swapping of two integer variables using third variable.
*5. Write a program for swapping of two integer variables Without using third variable.
*6. Write a program to print last digit of a given number.
*7. Write a program to calculate Compound Interest
*8. Write a program to swap two numbers.

#1. Write a program to find out square of given number
#2. Write a program to find out area of circle
#3. Write a program accept 5 subject marks (Hint P=67, C=87, M=90, H=98, E=88) and calculate total marks and percentage.
#4 Write a program accepts three numbers from user and calculate average of given three numbers.
#5. Write a program to accepts an amount in rupees (Hint Rs4567) and find out how many currency of Rs 2000 required. Also find remaining amount.
#6. Write a program to find power of a number using pow function
#7. Write a program to convert temperature from degree Fahrenheit to Celsius.
#8. Write a program to convert days into years, weeks and months.

#9. Write a program to flip bits of a binary number using bitwise operator.

#10. Find sum of first , third and fifth digit of 6 digit number.

Section 2 – Flow Control (Conditional Statements)

- If
- If-Else
- Else-If
- Nested If-Else
- Ternary operator
- Switch

*1. Write a program to accepts a number from user and check given number is even or odd.

*2. Write a program to accepts two numbers from user and calculates first no is divisible by second or not.

*3. Write a program to accepts three numbers from user and calculate biggest number out of three numbers.

*4. Write a program to calculate whether character is in lowercase or uppercase.

*5. Write a program to input basic salary of an employee and calculate its Gross salary according to following rules:

Basic Salary <= 10000 : HRA = 20%, DA = 80%
Basic Salary <= 20000 : HRA = 25%, DA = 90%
Basic Salary > 20000 : HRA = 30%, DA = 95%
Gross Salary = Basic Salary + HRA + DA

*6. Write a program to show day of week according to user input by using switch case.
*7. Write a program to perform all arithmetic operations according to user choice (for ex-for addition press '+'...) by using switch case.
*8. Write a program to find maximum between two numbers.
*9. Write a program to find maximum between three numbers using if-else and ternary operator.
*10.Write a program to calculate sum of digits of a number of three digit number using if-else
*11. Write a "Bonus Distribution Program" using logical operators. Bonus will be given to all those employees who have salary less than 20000 and tenure is more than 3 years.
*12 Write a code (using nested switch case) to suggest a diet plan (calories) to a consumer on behalf of inputs(gender and food time).

#1. Write a program that accepts the age of person, find out the person is eligible for voting or not.

#2. Write a program that accepts a number from user and find whether it is positive or negative or zero.

#3. Write a program to calculate whether year is leap year or not.

#4. Write a program that accepts five subjects 'marks from user and calculate the total marks then calculate Percentage. Display message according to following condition:

Percentage >=60 then print message Grade A
Percentage >=50 then print message Grade B
Percentage >= 40 then print message Grade C
Percentage < 40 then print message Grade D
#5. Write a program for generating electricity Bill. Accept last month unit and current month unit from user, then calculate and print bill amount according to following condition:
0-150 charges 4 rs/unit
151-300 charges 6 rs/unit
301-500 charges 8rs/unit
>500 charges 10rs/unit
#6. Write a program to show name of month . Ask user to enter between 1 and 12. Use switch case.
#7. Write a program that accepts a character and check given character is vowel or not by using switch case.
#8. Write a program to check whether a number is even or odd using switch case.
#9. Write a program to find the greatest of four numbers entered by the user.
#10. Write a program to calculate the income tax of an employee.
The tax slabs according to annual salary are :
upto rs.300000 tax is 0%
from rs.300000 to rs. 500000 tax is 10%
from rs.500000 to rs. 1000000 tax is 15%
more than 100000 tax is 20%
Note: 250000 is exempted from tax criteria
#11. Write a code for call center (using nested switch case). E,g, 1 for prepaid, 2 for post paid. If 1 selected then show all the options for prepaid plans

Section 3 – Flow Control(Loops)

- While Loop
- Repeat-While Loop
- Break & Continue
- For Loop

*1. Write a program to print "Code Better" five times by using loop.
*2. Write a program to print n natural number.
*3. Take any ten numbers from user and print sum and average of these numbers.
*4. Take any ten numbers from user and print sum and average of positive numbers.
*5. Take the numbers from user (until ten +ve numbers entered by the user), and print sum and average of these numbers.
*6. Write a program to calculate factorial of a given number.
*7.Write a program to calculate sum of digits of a number.
*8. Write a program to find out reverse of a given number.

*9. Write a program that accepts a number from user and check given number is Armstrong number or not.

*10. Write a program to find LCM of two numbers.

.e.g. LCM of 4 and 6 is 12

*11. Write a program to find HCF of two numbers.

.e.g. HCF of 16 and 24 is 8

*12. Write a program that accepts a number from user and check given number is prime number or not.

*13. Print Fibonacci series unto n terms 0,1,1,2,3,5,8,……

*14. Write a program to print given below patterns:

* * * * * * * * * * * * * * *	* * * * * * * * * *	A AB ABC ABCD ABCDE

*
* *
* * *
* * * * *
* * * * * * *
	* * * * *
	* * * *
	* * *
	* *
	*
		1 2 3 4 5
		1 2 3 4 5
		1 2 3 4 5
		1 2 3 4 5
		1 2 3 4 5

#1. Write a program to calculate square of numbers between 1-10

#2. Write a program to calculate cube of numbers between m and n. Ask user to enter value of m and n.

#3. Write a program to print table of any given numbers. . e.g table of 5 is 5, 10, 15,…,50

#4. Write a program that accepts a number from user and check given number is palindrome number or not. e.g palindrome number is 16761.

#5. Write a program that accepts a number from user calculate factor of a given number.

.e.g. factors of 12 are 1,2,3,4,6,12

#6. Write a program that accepts a number from user check given number is perfect number or not. A perfect number is whose sum of factors is wise of that numbers. e.g. factor of 6 are 1,2,3,6 then sum of 1+2+3+6=12

#7. Write a program to accept N number from user and show how many number are even or odd.

#8. Write a program to accept N number from user and check and print only Prime numbers.

#9. Write a program to accept N number from user and check and print only Armstrong numbers.

#10. Write a program to accept N number from user and check and print only palindrome numbers.

#11. Write a program to calculate sum of given series: 1-2+3-4+5-6+7-8……n.
#12. Write a program to calculate sum of given series: x + x^2 + x^3 + …….+x^n

#13. Write a program to print given below patterns:

1 12 123 1234	5 54 543 5432 54321	5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1	1 1 2 1 2 3 1 2 3 4 1 2 3 4 5
1 1 1 1 2 1 1 3 3 1 1 4 6 4 1	* * * * * * * * * * * * * * * *	* *	* *

#14. Print first letter of your name using start pattern

Section 4 - Functions

Function Types, Parameters, Declaration
Call by value & Call by reference
Scope, Visibility & Lifetime of Variable
Recursion

*1. Write a program to find cube of any number using function.
*2. Write a program to check whether a number is even or odd using functions.

*3. Write a program to find sum of digits of a given number using recursion.

*4. Write a program to check whether a number is palindrome or not using recursion.

*5. Create a function that return factorial of a number

*6. Create a function that return maximum element from array

#1. Write a program to check the prime number using function with argument and no return type.

#2. Write a program to print all even or odd numbers in given range using recursion.

#3. Write a program to find LCM of two numbers using recursion.

#4. Write a program to print all natural numbers between 1 to n using recursion.

Section 5 – Array

- - Array Concept, Declaration and Initialization, length
- One Dimension Arrays
- Two Dimention Arrays
- Zagged- Array

*1. Write a program to read and print elements of array.

*2. Write a program to find sum of all array elements

*3. Write a program to find maximum and minimum element in an array

*4. Write a program to insert an element in an array.

*5. Write a program to add two matrices.

*6. Write a program to search an element in an array.

*7. Write a program to sort an array .

*8 Write a program to reverse elements of an array

#1. Write a program to count total number of even and odd elements in an array.

#2. Write a program to copy all elements from an array to another array.

#3. Write a program to count total number of duplicate elements in an array.

#4. Write a program to merge two array to third array.

#5. Write a program to sort array elements in ascending or descending order.

#6. Write a program to multiply two matrices.

#7. Write a program to check whether two matrices are equal or not.

#8 Remove all duplicate occurring elements from array.

#9 Write a program to print words representation of entered number. e.g. if entered number is 245983 then result should be Two Lac Forty Five Thousand Nine Hundred Eighty Three

#10. Ask user to enter any four numbers between 1 to 9 and print all numbers made using combination of these four number. Do not repeat any digit in the same number.

#11. Find 2nd highest number from and 2nd minimum from array of n elements.

#12. Create a function that accept two Int arrays and return sum of all elements of both arrays

Section 6 - Object Oriented Programming

- Object oriented programming concepts- Object, Classes
- Polymorphism –static and dynamic
- Java-Class declaration, Compilation, Loading, Execution
- Java-Object References, Object allocation and Heap
- Instance Variable, Class Variable
- Instance Method, Class Method
-Constuctor-Default and Parameterised
- this keyword
- Static block,default initialization block
- Polymorphism and method overloading
- Methods with objects as arguments
- Has-A relationship
- Array of Objects, For Each Loop
- Method with Array as argument
- Variable Argument Methods
- Inner Class- static,member,local
- Class , object, reference, methods
- constructor , this keyword
- Static variable
- Array of objects, HAS-A relation
— GC and memory management

*1. Create a class Rectangle with three data member (length, breadth & area).
Now also create method members
• inputValue() - to take input for length and breadth from user.
• calculateArea() - to calculate area of rectangle.
• showArea() - to display the area of rectangle.

*2. Create a class DemoArithmetic with three data member that are firstNo, secondNo and result .
In this class also create following method:
a. inputValue() - to take input value from user for firstNo and secondNo
b. addition() - to perform addition operation and store in result var
c. substraction() - to perform substraction operation and store in result var
d. multiplication() - to perform multiplication operation and store in result var
e. divide() - to perform divide operation and store in result var
f. show() - display value of result variable.

*3. Create a class Student with data member(stuName, stuId, stuPercentage) to store the information of student And also create following method:
a. inputStuInformation() - to take infromation about student from user
b. outputStuInformation() - to display information of student

*4. Create a class Addition with member function addition() with following given argument
a. three float
b array of integer
c two integer

#1. Create a class Student with data member (stuId ,stuName ,stuPer)) by using following properties:
a. Only parameterized constructor;

b. ShowStuInformation() method - display the information of student.
c. Create three student object and call ShowStuInformation method

#2. Create class Employee with variables name, id, salary.
- Create default and parameterized constructor.
- Create constructor for id and salary only.
- Use this keyword in all initializers

#3. Create class Farmer with variables (name, crop, earning).
- Create default and perameterized constructor
- Create display function
- Compare earnings of two Farmers and print name and crop of farmer who earns more.

#4. Create class Circle with variable radius.
- Create default init to set radius=1.
- Create parameterized constructor.
- Create display function to display radius and area.
- Create function isBiggerThan(other : Circle) -> Bool
- Create Two objects and call all the above functions

#5. Create class BankAccount with data members accountNo, name, balance.
- Create display() method.
- Create another method deposit(amount:Float) which add amount value to balance
- Create withdraw(amount:Float) method which subtract amount from balance.
- Store 5 BankAccount Information in an Array and display all records. Ask user to select

choice from below and perform operation
1. Display account detail by account number
2. Deposit amount in account by account number
3. Withdraw amount in account by account number
4. Delete account by account number.

Section 7 - Inheritance , Abstract Class, Interface

- Introduction, Types,concept,declaration
- Multilevel inheritance and Constructor chaining
- Extends Keyword
- Constructor in inheritance, super keyword
- Method overriding
- Abstract class, Interface
- Runtime Polymorphism
- Instanceof and final keyword usage
- References of super types, Dynamic Polymorphism

*1. Create class Student with variable (name, course, dateOfBirth)
- create SchoolStudent class with variable (fees) and inherit class Student
- create CollegeStudent with variable(sem and fees) and inherit class Student
- create class PGStudent with variable(mainSubject, percent) and inherit CollegeStudent
- add default and parameterized constructor in all classes

*2. Create class Employee(id, name, salary) and add display function

- create class PartTimeEmployee with variable(hoursWorked , hourlyRate) and inherit - class Employee. Add calculateSalary() function to calculate salary as salary = hourlyRate * hoursWorked

- create class FullTimeEmployee with variables(basic, da, pf) and Add calculateSalary() function to calculate salary as salary = basic + da - pf

- add default and parameterized constructor in all classes

- Create a global function getTax(emp:Employee) which accept Employee object in argument and calculate and print income tax on employee salary(12.5% of salary)

- Also try to pass Objects of PartTimeEmployee and FullTimeEmployee to getTax() method

*3. Create class BankAccount with variables (accountNo, balance).

- Add default and parameterized init.
- Add display method
- Add depositAmount(a:Float) method to increase balance
- Add withdrawAmount(a:FLoat) to subtract from balance

Create class SavingAccount with variable(customerName,homeAddress) and Inherit class BankAccount.

- Add static variable minimumBalannce.
- Add display method
- override withdrawAmount(a:Float) to check minimumBalance

Create class CheckingAccount with variable(shopName, shopAddress) and Inherit class BankAccount.

- Add static variable overDraftLimit.
- Add display method
- override withdrawAmount(a:Float) to check overDraftLimit

Create an Array of BankAccount type and append 3 objects of SavingAccount type and 4 objects of CheckingAccount Type with default data. Display all account information on screen

#1. Create a class Bird with properties – name, habitat, food, feather color. Create Abstract Class FlyingStyle, which has a method fly (). Inherit this class to Bird class and write code for the fly method.

#2. Create interface Taxable with variables taxRate{get}, and function getTax()->Float

- create class Employee(id,name,salary) add display function. Implement Taxable interface. Decide income taxRate 30% for salary above 10 lac, 20% for salary between 5 and 10 lac and 6% for salary below 5 lac
- create class Restaurent (restaurent name, Bill amount). add display and input function. decide GST 10% and implement Taxable protocol to getTax()
- create 2 object of Employee and 3 Obj of RestaurentBill and print total amount of tax on all 5 objects
- Create Global function that accept reference of type Taxable and print 5% of refund on tax applied.

#3

- Create an Abstract class Processor with int member variable data and method showData to display data value. Create abstract method process() to define processing of member data.
- Create a class Factorial using abstract class Processor to calculate and print factorial of a number by overriding the process method.
- Create a class Circle using abstract class Processor to calculate and print area of a circle by overriding the process method

Ask user to enter choice (factorial or circle area). Also ask data to work upon; Use Processor class reference to achieve this mechanism

#4 Explain the importance of toString() and equals() method of the Object class and override them on class Employee(empId,name,salary).

- Create class for main method(say XYZ),and accept five employees information and store in an array. Also ensure if entered empId already exist or not (use equals method).
- Display all employee info using toString method;

Section 8 - Packages, Jar file, String and Wrapper classes

- Organising classes in packages
- Access Protection – private, default, get, set methods
- Access Protection – protected, public
- Sub Packages
- Compiling packages, Making and running.jar
- Java. Lang package, String class and its methods
- Object class, toString, isEqual, hashCode
- Wrapper Classes – Integer, Float Boolean etc.
- CommandLine Arguments

*1. Create a following class/package structure in your application. Filled squares are packages and empty circles are classes.

- campus.data
  - o Person (name, age, address)
- campus.academics
  - o Student(rollno, branch, semester) inherits Person
  - o Faculty(facultyId, name, salary, branch) inherits Person
- campus
  - o CampusApp- this class contains **main** method to accept 5 Faculty information and print total of salaries of all faculties.

#1. Add following package to above structure

- campus.accounts
  - o Fees(Student, amount, paymentDate)
- campus -
  - o CampusApp- this class contains **main** method to accept 4 Students information with their fees for second semester and print total fees collected from all 4 students.

Take your assumptions regarding subject, branch and student data. Create this application as a runnable jar file.

Section 9 - Exception Handling

- Concept
- Role of JVM

- Types of Exception
- Class Hierarchy
- Try, catch, finally – common exceptions
- throw, throws
- Custom Exception – checked and un-checked
- Handling Multiple Exceptions

*1. Store name of weekdays in an array (starting from “Sunday” at 0 index). Ask day position fromuser and print day name. Handle array index out of bound exception and give proper message if user enter day index outside range (0-6).

#1. Create a class Voter(voterId, name, age) with parameterized constructor. The parameterized constructor should throw a checked exception if age is less than 18. The message of exception is **“invalid age for voter ”**

#2. Create Interface **StudentFee** and declare following method.

getFee() throws InvalidFeeException. This method ask fees from user and throws exception if user enters invalid or negative fees

Create class **Student** with members (name, fees) and implement the **StudentFee** Interface.

Section 10 - Multi Threading

- Concept of Multi Threading- OS, Process, Application, Thread
- Thread Class- Thread creation, execution
- Runnable Interface – Thread creation, execution
- Sleep, join, interrupt
-Thread Synchronisation
- Thread name, priority and thread states

*1. Create a **Thread** class to print following star(*) pattern on screen with delay of 1 second between each * print. Number of lines in the pattern should be passed to the constructorof Thread class.

```
* * * * *
 * * * *
* * *
* *
*
```

Use this class in main method and ask user to enter number of lines to print.

#1. Create a class that checks whether a given number is prime or not using Runnable interface

#2. Develop a stop watch

Section 11 - Java IO

- Java IO – Concept of streams
- Input Streams – text and binary, standard input
- Output Stream – text and binary, standard output
- Text File Reading Writing
- Binary File Reading Writing
- DataInputStream/DataOutputStream
- ObjectInputStream/ObjectOutputStream
- Searialization and Other Streams

1. Write a program to count how many times character ‘t’ occurs in a file.
2. Write a program to count no of words in a text file and average word size.
3. Write a program to count number of bytes in a image file(jpeg/png/gif). Also find how much time it will take to upload the file on server if internet speed is 256 bps(bits per second).
4. Write a program to store your shopping details in a binary file(**shopping.dat**) with information itemName, price, quantity. (Use ObjectOutputStream to store Item class object).
5. Write a program to read data from **shopping.dat** file creted in above problem and find total money spent on all shopping items. . (Use ObjectInputStream to read Item class object).

Section 12 – Generics

- Concept of Generics
- Generic Types
- Generic Methods
- Bounded Type Parameters
- Generic Type Inheritance and Subtypes
- Wild cards

*1. Implement Swapping of two values using Generics

#1. Implement Queue operations using Generics

Section 13 - Strings

- Declaring Strings in
- Strings Input and Output functions
- String Comparison
- String Functions

*1. Write a program to find length of a string.

*2. Write a program to take a two as an input from the user . Confirm that the strings are equal.

*3. Write a program to find total number of alphabets, digits or special character in a string.

*4. Write a program to convert lowercase string to uppercase.

*5. Write a program to find reverse of a string.

#1. Write a program to convert string into lowercase without any library function.

#2. Write a program to accept a string and check if it is palindrome or not?

#3. Write a program to count total number of vowels and consonants in a string.

#4. Write a program to find first occurrence of a character in a given string.

#5. Write a program to toggle case of each character of a string.

Section 14 - Collections ArrayList, Map , Set

- Concept of Collection, Class Hierarchy
- List and Subclasses
- Queue and Subclasses
- Set and Subclasses
- Map and Subclasses
- Comparator, Comparable
- Collection algorithms

*1. Create a map to store student rollno (key) and name(value). store 5 pairs by default

*2. Create a map to store name_of_course (key) and duration_in_month(value). store 5 pairs by default

*3. Declare Class to store information about Book Title, Price, Auther, pageCount. Store information of any 4 books in an ArrayList and print as below

```
Title          Price   Auther        Pages
____________________________________________

Let US C       450     Kanitkar      338

.....................................
```

*4. Create a Map to store name of cricket teams according to their world rankings

*5. Create a set of different movie star names. Perform these operations on the set

a. Display all the elements of the set

b. Remove the element at position 3

c. Check if a given name exist in the set

d. Display the total number of names in the set

#1. Create a Map to store fruitname (key) and fruit price(value). store 5 pairs by default

#2. Create a Map to store name_of_course (key) and duration_in_month(value). store 5 pairs by default

#3. Create Map to store name of states and 3 cities of each state

#4. Create a set of 10 numbers and perform these operations on the set

a. Sort the elements in ascending and descending order

b. Find the average of the elements

c. Identify the highest and the smallest element

d. Remove the smallest element in the set

f. Print the contents of the set

Section 15 - Mini Project Ideas

1. Create a Student Management System to store , delete, update list records of student. Store rollno, name, course, semester , percentage

2. Create BankAccount Management App to store , delete, update, list, deposit, withdraw, search records of Bank Account. Store accNo, customer name, balance, account type

3. Create CodeBetter enquiry system to store, delete, update, list, search enquiry details. Store enquiry details like candidate name, contact, address, course selected, course fee